Resilient Approximation-Based Distributed Nonconvex Optimization

Yilin Zhang, Zhiyu He, and Jianping He

Abstract— **There has been an approximation-based distributed optimization algorithm that solves univariate nonconvex problems to arbitrary precision. The key idea is to construct approximations of local objectives and address a more structured approximate version of the problem. By representing diverse local objectives with compressed coefficients vectors, such algorithms enjoy gradient-free iterations but face severe security issues when adversaries occur. In this paper, we propose a resilient approximation-based distributed nonconvex optimization algorithm termed R-ADOA to defend attacks from malicious nodes. First, errors caused by adversaries are quantified and unified as the perturbation of coefficient vectors of approximations. Next, we propose a filtering mechanism and resilient stopping mechanism to limit errors arising in consensus-based iterations. Finally, an upper bound of the deviations of the obtained solutions from optimal solutions is given based on the eigenvalue perturbation theory of matrices. Numerical experiments are provided to illustrate the effectiveness of our algorithm. Compared to existing resilient distributed optimization algorithms, R-ADOA addresses nonconvex problems, converges exponentially fast, and contains explicit bounds for the deviations of solutions.**

I. Introduction

Distributed optimization has become a fundamental building block of a variety of applications related to network systems, including robotic coordination [1], federated learning [2], and energy management [3]. In the past years, methods like gradient tracking and dual decomposition have been introduced to efficiently solve distributed convex optimization problems [4], [5]. Nowadays, the challenging nonconvex problems draw wide interest because of their generality and practical importance. Several techniques are utilized to tackle distributed non-convex optimization, including stochastic gradient descent, successive convex approximation, and proximal methods [6]–[9]. Notably, approximation-based optimization algorithms receive increasing attention due to their high efficiency. First, these algorithms compress local objective functions into their approximation proxies and use consensus-based gradient-free iterations, which effectively reduces the costs in communication. Second, approximations with nice analytic properties are easier to optimize when compared to general objective functions.

To achieve network-wide agreement on optimal solutions, a majority of the above algorithms rely on the assumptions of trustworthy neighbors and accurate computations. However, since distributed algorithms are extensively applied in large-scale networks, there is a high risk of these algorithms being disrupted by cumulative noises or being attacked by adversarial nodes [10]. This issue is especially critical for approximation-based algorithms. First, vectors rather than scalar variables are exchanged between neighbors, which implies a larger space for adversarial attacks. Second, slight perturbations to highly compressed proxies may cause deviations in representing objective functions. Finally, the required local stopping mechanism can lose effect in face of adversaries. Therefore, it is necessary to consider resilient designs that mitigate errors happened in distributed optimization [11].

Research on distributed algorithms considering security factors has been conducted both on average consensus and distributed optimization. Studies in these areas mainly consist of two aspects: i) normal nodes adopt a detection algorithm to identify adversarial nodes, and ii) normal nodes employ a filtering mechanism that drops extreme values. For the first aspect, researchers set up different attack models and create corresponding detection mechanisms [12]–[14]. Moreover, others require local nodes to utilize two-hop information to detect adversarial nodes [15], [16]. As for the other aspect, one family of algorithms is based on Mean-Subsequence Reduced (MSR) methods, where the detailed investigations can be found in [17]–[19]. Based on the MSR method, various resilient distributed optimization algorithms are proposed to solve univariate convex problems [20]–[24]. These algorithms solve univariate convex problems with sublinear rates and ensure the convergence to minimizers of convex combinations of local objectives [12], [21].

The approximation-based distributed optimization algorithm has significant differences in structures. Hence, existing design and analysis for resilient distributed algorithms cannot be directly applied. For approximation-based distributed algorithms, attacks happen in the first two phases but the errors can be unified as a perturbation of vectors. To enhance the security and robustness of the approximation-based methods, we adopt a resilient average consensus mechanism to constrain errors. We combine the standard MSR algorithm with a newly defined local information set to form a W-MSR algorithm. A resilient stopping mechanism is introduced to terminate the consensus process when local proxies are close enough. Next, based on the proxy acquired after consensus-based iterations, an approximation of the global objective function is restored, and the minimum value of this approximation is obtained locally. In addition, we classify the inevitable interference of adversaries as two types of errors and derive upper bounds of the performance loss due to these errors. The main contributions are summarized as follows.

The authors are with the Department of Automation, Shanghai Jiao Tong University, the Key Laboratory of System Control and Information Processing, Ministry of Education of China, and Shanghai Engineering Research Center of Intelligent Control and Management, Shanghai, 200240, China. E-mail address: {yilinsjtu, hzy970920, jphe}@sjtu.edu.cn. This work was supported by the NSF of China under Grant 61973218.

TABLE I: Important Notations

Symbol	Definition
$\mathcal{G}$	the network graph
D	the diameter of $\mathcal{G}$
U	an upper bound on D known to all the agents
W^t	the weight matrix at time t
m	the degree of local approximation
$\mathcal{N}_i^{(r)}$	neighbor set of node i at r-th dimension
ϕ_j	the j-th dimension type-I error of node i
$\delta_i(j)$	the j-th dimension type-II error of node i
$p_i(t)$	the approximation proxy at iteration t
$p_i^{(j)}(t)$	the j-th dimension coefficient at node i at time t
$\hat{p}_i^K(x)$	the approximation polynomial recovered by node i
$p_i^K(x)$	the approximation polynomial to global objective function
K	a large time when all normal nodes reach consensus

- We propose a Resilient Approximation-Based Distributed Non-convex Optimization Algorithm (R-ADOA) that solves the univariate non-convex optimization problem against malicious attacks. It maintains the efficiency of approximation-based algorithms and owns the resilience against perturbation.
- We classify the errors in R-ADOA as two types, i.e., approximation error and communication error. We use the weighted orthogonality of Chebyshev polynomials to characterize approximation errors as a function of the perturbation imposed by adversaries. We provide explicit formulas of communication errors caused by W-MSR consensus algorithms in each dimension.
- We use the eigenvalue perturbation theory of matrices for sensitivity analysis and derive an upper bound of the distance between the perturbed optimal solution and the original optimal value. Numerical evaluations are conducted to show relative errors of our algorithm.

The remainder of this paper is organized as follows. Section II provides some preliminaries and formally defines the problem. Section III presents the design of the proposed resilient nonconvex optimization algorithm. In Section IV, we analyze the convergence and accuracy of the proposed algorithm. Numerical evaluations are given in Section V. Finally, Section VI concludes the paper and discusses future directions. We summarize notations in Table I for reference.

II. Preliminaries and Problem Formulation

A. Models of Networks and Adversaries

A network is represented by an undirected connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, ..., N\}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. Nodes i and j can communicate with each other if and only if $(i, j) \in \mathcal{E}$. Let $\mathcal{N}_i$ be the set of neighboring nodes that can exchange information with node i. We use $\mathcal{V}_n$ and $\mathcal{V}_a$ to denote the sets of normal nodes and adversarial nodes, respectively.

In this paper, we consider malicious adversaries that send the same manipulated values to their neighbors. This type of adversaries is widely investigated in the field of cybersecurity and federated learning [11]. Typical examples include DDoS, data injections [12] and errors caused by accidental faults.

Assumption 1. *There are in total g malicious nodes in $\mathcal{G}$, and $g \leq \frac{N-1}{2}$.*

Assumption 2. *The network $\mathcal{G}$ is (g+1, g+1)-robust.*

Assumption 3. *The local objective function $f_i(x)$ is Lipschitz continuous.*

The bound on g to ensure the convergence of distributed algorithms in face of adversaries is widely assumed [23]. The assumption on the network topology is necessary for the normal nodes adopting a W-MSR algorithm to reach consensus. More detailed explanations of $(g+1, g+1)$-robust can be found in [19].

B. Preliminaries

To facilitate the design and analysis, we provide some basic knowledge as follows.

- *Approximation Theory*

The central idea of approximation is to express a function as a finite combination of the basis for the function space. For example, the Fourier expansion for a function relies on the Fourier basis, i.e., sinusoidal functions. Let $F_j(x)$ be the j-th basis function and m_i be the degree of approximation. Then, the approximation function of degree m_i for $f_i(x)$ is

$$p_i^{(m_i)}(x) = \sum_{j=0}^{m_i} p_i^{(j)} F_j(x), \quad x \in [a, b]. \tag{1}$$

Therefore, $f_i(x)$ can be closely represented by a vector of coefficients $p_i = [p_i^{(0)}, p_i^{(1)}, \ldots, p_i^{(m_i)}]^\top$. We term this vector the approximation proxy of $f_i(x)$. The degree of approximation should be decided in advance so that every node can have a proxy in the same dimension length. However, this may lead to larger errors or unnecessary calculation consumption, but it will help guarantee safety instead.

In this paper, we use the Chebyshev basis for its simplicity and rich supporting theories [25], [26]. Each node constructs a local Chebyshev proxy by using the adaptive Chebyshev interpolation method [26]. As the degree of approximation is fixed beforehand as m, each node will first calculate $m+1$ interpolation points of $f_i(x)$ according to

$$x_k = \frac{b-a}{2}\cos\left(\frac{k\pi}{m}\right) + \frac{a+b}{2}, \quad f_k = f_i(x_k), \tag{2}$$

with $k = 0, 1, ..., m$. Next, based on $m+1$ interpolation points $\{(x_k, f_k)\}$, the coefficients of the Chebyshev proxy are

$$p_i^{(j)} = \frac{1}{m}(f_0 + f_m \cos(j\pi)) + \frac{2}{m}\sum_{k=1}^{m-1} f_k \cos\left(\frac{jk\pi}{m}\right), \tag{3}$$

where $j = 0, 1, ...m$ [27]. The proxy is represented by a coefficient vector $p_i = [p_i^{(0)}, p_i^{(1)}, \ldots, p_i^{(m)}]^\top$, and the approximation polynomial can be constructed by (1). More detailed analysis can be found in [26].

- *Average Consensus*

Suppose that in $\mathcal{G}$, each node i holds an initial value $x_i^0 \in \mathbb{R}$. The goal of average consensus is to enable agents to agree on the average of initial values $\bar{x} = \sum_{i=1}^N x_i^0 / N$ via local communication and computations. The update rule for each node i is as follows

$$x_i^{t+1} = W_{ii}^t x_i^t + \sum_{j \in \mathcal{N}_i} W_{ij}^t x_j^t,$$

where $\mathcal{N}_i$ is the set of neighbors of node i. In a static graph, the coefficient matrix W^t is designed to be doubly stochastic to guarantee that x_i^t exponentially converges to $\bar{x}$. In that case, we can set W^t by using constant weights, local degree weights, or Metropolis weights [28], [29]. For time-varying graphs, further assumption on the topology, e.g., *B-strongly-connectivity*, is required to ensure convergence [29].

C. Problem Formulation

Suppose that the network $\mathcal{G}$ involves N nodes, each of which holds a possibly non-convex local objective function $f_i(x) : \mathbb{R} \longmapsto \mathbb{R}$. There are n normal nodes and g adversarial nodes, such that $n + g = N$. Without loss of generality, we assume that the first n nodes are normal. The goal is to enable normal nodes to collaboratively optimize a global objective function $f(x)$, which is the average of their local objectives, i.e.,

$$f^* = \min_{x \in X_R} f(x) = \min_{x \in X_R} \frac{1}{n} \sum_{i=1}^{n} f_i(x). \tag{4}$$

where X_R denotes the constraint set. In an approximation-based algorithm, $f(x)$ is closely approximated by a finite series $p_i^K(x)$, which is locally optimized by each agent i to obtain an estimate of the optimal value of problem (4). When adversarial nodes exist, the locally recovered approximation $\hat{p}_i^K(x)$ can be interpreted as adding perturbations on the coefficients of $p_i^K(x)$. In this case, we analyze the bounds on distances in terms of optimal points and optimal values, i.e.,

$$|x^* - \hat{x}^*| \quad \text{and} \quad \left| f^* - \hat{p}_i^K(\hat{x}^*) \right|, \tag{5}$$

where

$$x^* = \operatorname*{argmin}_{x \in X_C} p_i^K(x), \quad \hat{x}^* = \operatorname*{argmin}_{x \in X_C} \hat{p}_i^K(x). \tag{6}$$

III. Algorithm Design

In this section, we present the design of R-ADOA. The key idea of the proposed algorithm is as follows. First, each node constructs a Chebyshev proxy of its local objective function. Then, a W-MSR algorithm and a resilient stopping mechanism are applied to guarantee that normal nodes have their proxy reaching consensus. Finally, each node recovers an approximation of the global objective function and locally solves the polynomial optimization problem to obtain solutions. The whole algorithm is presented in Algorithm 1.

A. Algorithm Details

The algorithm details are introduced in this subsection. We start with the updating rule and resilient stopping mechanism. The updating rule is to define a filtering mechanism so that normal nodes use values that have been selected to update their local values. In this paper, the W-MSR mechanism proposed in [19] is adopted.

We first introduce the following notations to describe the filtering mechanism of W-MSR. We define the filtered set of $\mathcal{N}_i^{(r)}$ at iteration t as $\mathcal{N}_i^{(r)*}(t)$. The median of values of $\mathcal{N}_i^{(r)}$ is denoted as $M_i^{(r)}$. For simplicity, we define $p_{m_i}^{(r)}(t) = \min\{p_i^{(r)}(t), M_i^{(r)}\}$ and $p_{M_i}^{(r)}(t) = \max\{p_i^{(r)}(t), M_i^{(r)}\}$. Then we construct two sets that obtain locally constrained values and non-extreme values, respectively. First, denote $\mathcal{N}_{b_i}^{(r)}(t)$ as the set of values that are bounded by local value and the median of its neighbors. Second, let $\mathcal{N}_{w_i}^{(r)}(t)$ be values selected by MSR. The mathematical expression of $\mathcal{N}_{b_i}^{(r)}(t)$ and $\mathcal{N}_{w_i}^{(r)}(t)$ are as follows,

$$\begin{aligned} \mathcal{N}_{b_i}^{(r)}(t) &= \{j \in \mathcal{N}_i^{(r)} \mid p_{m_i}^{(r)}(t) \le p_l^{(r)}(t) \le p_{M_i}^{(r)}(t)\}, \\ \mathcal{N}_{w_i}^{(r)}(t) &= \mathcal{N}_i^{(r)}(t) \backslash \mathcal{N}_{m_i}^{(r)}(t) \backslash \mathcal{N}_{M_i}^{(r)}(t), \end{aligned} \tag{7}$$

where $\mathcal{N}_{m_i}^{(r)}(t)$ and $\mathcal{N}_{M_i}^{(r)}(t)$ represent the set of indices of g-smallest and g-largest values, respectively. Therefore, $\mathcal{N}_i^{(r)*}(t)$ satisfies

$$\mathcal{N}_i^{(r)*}(t) = \begin{cases} \{i\}, & \text{if } |\mathcal{N}_{w_i}^{(r)}(t)| = 0, \\ \mathcal{N}_{b_i}^{(r)}(t) \cup \mathcal{N}_{w_i}^{(r)}(t), & \text{otherwise.} \end{cases} \tag{8}$$

Actually, as we have Assumption 2 on network topology, the case where $\mathcal{N}_i^{(r)*}(t)$ has only one element will seldom happen. From this definition of the filtered set $\mathcal{N}_i^{(r)*}(t)$, we can see how W-MSR considers locally constrained values and non-extreme values. Once the set of values is found out, each node will update its local value by the following rule,

$$p_i^{(r)}(t+1) = \sum_{j \in \mathcal{N}_i^{(r)*}(t)} w_{ij}^t p_j^{(r)}(t). \tag{9}$$

Remark 1. *Though transmitting m-dimensional coefficient vectors may lead to increased burden in each round of communication, approximation-based methods generally enjoy reduced costs in total communication because of their rapid convergence and typically moderate degrees m [8], [25].*

Next, we establish the convergence of normal nodes to the approximate average of their initial states. The key point here is properly selecting coefficient w_{ij}^t to guarantee the matrix W^t is row-stochastic, which means that $\sum_{i=1}^{N} w_{ij}^t = 1$. As discussed before, our algorithm only makes use of values that are in the $\mathcal{N}_i^{(r)*}(t)$ set. Therefore, values lying out of the set have no contribution to the updating of local value and their corresponding coefficients are set to be zero. Precisely, the formula below gives the determined matrix W^t

$$w_{ij}^t = \begin{cases} \frac{1}{\left|\mathcal{N}_i^{(r)*}(t)\right|}, & j \in \mathcal{N}_i^{(j)*}(t), \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Finally, we propose the resilient stopping mechanism. To decide when to stop our consensus communication, we use the max/min consensus-based stopping mechanism mentioned in [8]. This mechanism adds two auxiliary variables $r_i(t)$ and $s_i(t)$ initialized as vector p_i tracking the maximum and minimum elements in each node. The information that

```
Algorithm 1: R-ADOA
Input: $\mathcal{G}$, $M$, $K$
Output: $x^*$, $p_i^K(x^*)$
for $i$=1:N do
    Node $i$ calculates $\{x_k\}$ and $\{f_k\}$ by (2).
    Node $i$ constructs Chebyshev proxy $p_i$ by (3).
    Initialize $p_i(0) = r_i(0) = s_i(0) = p_i$.
    for $t$=0:K do
        for $r$=1:M do
            // Node $i$ finds $\mathcal{N}_i^{(r)*}(t)$,$\mathcal{R}_i^{(j)*}(t)$
            and $\mathcal{S}_i^{(j)*}(t)$.
            Update $p_i^{(j)}(t)$ by (9).
            Update $r_i^{(j)}(t)$ and $s_i^{(j)}(t)$ by (11).
        end
        if $t$ Mod $U$==0 then
            $p_i(t) = r_i(t) = s_i(t)$.
            if condition (12) holds then
                $p_i(K) = p_i(t)$.
                break
            end
        end
    end
end
Node $i$ recovers $p_i^K(x)$ from $p_i(K)$.
Node $i$ gets $x^*$ and $p_i^K(x^*)$ by local computation.
return $x^*$, $p_i^K(x^*)$.
```

auxiliary variables carried spread the whole network in at most D rounds update, where D is the diameter of $\mathcal{G}$. Therefore, the auxiliary variables will be reinitialized periodically. In another word, each node should retain information about the diameter of the graph, which implies the stopping requires the following assumption.

Assumption 4. *Every node knows an upper bound U on D.*

Note that nodes can obtain such a bound by estimating D via the Extrema Propagation technique [30]. The aforementioned auxiliary variables are updated according to

$$r_i^{(j)}(t+1) = \max_{k\in\mathcal{R}_i^{(j)*}(t)} r_k^{(j)}(t),\ s_i^{(j)}(t+1) = \min_{k\in\mathcal{S}_i^{(j)*}(t)} s_k^{(j)}(t), \tag{11}$$

where $\mathcal{R}_i^{(j)*}(t)$ and $\mathcal{S}_i^{(j)*}(t)$ are defined in the same way as the definition of set (8). These auxiliary variables are reinitialized every U-th round. For more details on max/min stopping mechanism, the readers can refer to [31].

B. Local Polynomial Optimization

After each normal node reaches consensus on approximation proxy, they recover the approximation polynomial $p_i^K(x)$ from equation (1). The optimization problem can be solved by comparing values of $p_i^K(x)$ on boundaries and extreme points which are roots of $\frac{dp_i^K(x)}{dx}$. Collectively, these points are referred to as critical points and we use X_C to denote the set of all critical points.

IV. Performance Analysis

In this section, we characterize the performance of R-ADOA under malicious attacks. There exist two types of errors in the whole process. One is the local approximation error, while the other is the global communication error. Both errors will cause perturbation of coefficients and hence disturb the optimization. First, we elaborate that by using the W-MSR algorithm, the stopping mechanism is valid and the average consensus can be reached. Second, we analyze the characteristics of these two types of errors. Finally, we study how the aforementioned errors will influence the solution accuracy and provide a bound on the performance loss due to these errors.

A. Two Important Lemmas

For the first lemma, we show that under Assumptions 1-2, the W-MSR algorithm will guarantee normal nodes reaching consensus. After adopting filtering mechanism, the updating rule of normal nodes can be seen as updating rule of a time-variant directed graph. A theorem in [32] illustrates sufficient conditions for the convergence of a sequence of row-stochastic weight matrices in the time-variant directed graph. The sufficient conditions are i) each node has a self-loop, ii) for $j \in \mathcal{N}_i^{(r)*}(t)$, $w_{ij}^t > \alpha > 0$, and iii) in a fixed time interval δt, there exists a globally reachable node. Later in [19], a sufficient and necessary condition for normal nodes which adopt the W-MSR algorithm reaching consensus on undirected graph is proposed. Let $\Phi(t) = W^t W^{t-1} \cdots W^0$ be the product of determined matrices before time t.

Lemma 1 ([19])**.** *Suppose Assumptions 1-2 hold and normal nodes follow W-MSR algorithms, then there exists a stochastic vector $\omega \in \mathbb{R}^N$ such that $\lim_{t\to\infty} \Phi(t) = \mathbf{1}_N \omega^\top$.*

The second lemma is about the eigenvalue perturbation theory of a matrix. It states the change of eigenvalues if a perturbation is imposed on the matrix. Suppose we have a square matrix $\hat{M}_C$ and add a perturbation E on it as $M_C = \hat{M}_C + E$. Then, we have the following lemma.

Lemma 2 ([33])**.** *If $\hat{M}_C$ has the Jordan canonical form $X^{-1}(\hat{M}_C)X = \operatorname{diag}\left(J_{n_1}(\lambda_1), J_{n_2}(\lambda_2), \ldots, J_{n_k}(\lambda_k)\right)$. Then for any $\mu \in \sigma(M_C)$ there exists $\lambda_j \in \sigma(\hat{M}_C)$ such that*

$$\left|\mu - \lambda_j\right| \le 1/g_{n_j}(1/\theta) \le \max\left\{n_j\theta, \left(n_j\theta\right)^{1/n_j}\right\}$$

holds with $\theta = \left\|X^{-1}EX\right\|_2$, $\sum_{j=1}^{k} n_j = m-1$. $\sigma(M_C)$ is the set of eigenvalues of matrix M_C and $g_\alpha(c)$ is the unique non-negative real zero of equation $\phi_\alpha(x) = \sum_{l=1}^{\alpha} x^l = c(c \ge 0)$.

B. Convergence of R-ADOA

In this section, we propose a theorem that states how the stopping mechanism helps to decide when to stop the consensus phase. The initial vector at k-th dimension is defined as $\Omega(k) = [p_1^{(k)}(0), p_2^{(k)}(0), \ldots, p_N^{(k)}(0)]^\top$ with $p_i^{(k)}(0) = p_i^{(k)}$. Then by Lemma 2, we have $\{\omega_1^\top\Omega(1), \ldots, \omega_m^\top\Omega(m)\}$ is the final result that normal nodes converge to.

Theorem 1. *Suppose that Assumptions 1-4 hold. R-ADOA ensures that there exists a time $t_o > 0$, such that for any* *specified* $\delta > 0, \forall i \in \mathcal{V}_n, \forall 0 \le j \le m,$

$$\begin{cases} |r_i^{(j)}(t_o+1) - s_i^{(j)}(t_o+1)| \le \delta, \\ |r_i^{(j)}(t_o+1) - \max\limits_{k\in \mathcal{N}_i^{(j)*}(t)} p_k^{(j)}(t_o)| \le \delta, \\ |s_i^{(j)}(t_o+1) - \min\limits_{l\in \mathcal{N}_i^{(j)*}(t)} p_l^{(j)}(t_o)| \le \delta, \end{cases} \tag{12}$$

where δ *is a given small constant. Meanwhile, we have*

$$p_1(t) = \cdots = p_n(t) = \{\omega_1^\top \Omega(1), \ldots, \omega_m^\top \Omega(m)\}$$

holds for $\forall t > t_o$*, where* ω_1 *to* ω_m *are stochastic vectors.*

Proof. See Appendix A. □

Therefore, by periodically checking if (12) is satisfied, each node will know when the consensus is reached.

C. Error Analysis of R-ADOA

In this section, we formulate the errors present in R-ADOA. There are two types of errors happening in the approximation phase and the consensus phase, respectively.

- *Case I: Approximation Error*

As the approximation is progressed individually, the adversarial nodes can only affect its own approximation. We denote the additional bias from objective function $f_i(x)$ as $\varphi_i(x)$ and the function after attacking is $\breve{f}_i(x)$. Correspondingly, their coefficients vectors will be p_i and $\breve{p}_i$. From (1), it is derived that $f_i(x) = p_i^{(m)}(x) + \varepsilon_i(x)$ and $\breve{f}_i(x) = \breve{p}_i^{(m)}(x) + \breve{\varepsilon}_i(x)$. Denote $\phi_i(j) = \breve{p}_i^{(j)} - p_i^{(j)}$ as the approximation error on j-th dimension of node i. The following theorem quantifies $\phi_i(j)$.

Theorem 2. *Suppose Assumptions 3 holds. With R-ADOA, the approximation error* $\phi_i(j)$ *is given by*

$$\phi_i(j) = \breve{p}_i^{(j)} - p_i^{(j)} = \alpha_j \int_{-1}^{1} \frac{h_i(t)\,T_j(t)}{\sqrt{1-t^2}}\mathrm{d}t, \tag{13}$$

where $i \in \mathcal{V}$,

$$\alpha_j = \begin{cases} 1/\pi, & j = 0, \\ 2/\pi, & 0 < j \le m, \end{cases} \tag{14}$$

and $h_i(t) = \varphi_i(x(t)) + \varepsilon_i(x(t)) - \breve{\varepsilon}_i(x(t))$*, in which* $x(t) = [(b-a)t + a + b]/2$ *for* $j = 0, 1, ..., m$.

Proof. See Appendix B. □

Though malicious nodes can manipulate the proxy arbitrarily in the approximation phase, we unify the change of coefficients as attackers imposing a perturbation function. For normal nodes, $\varphi(x) = 0$ and $\varepsilon_i(x) = \hat{\varepsilon}_i(x)$ because their approximations are not disturbed, which means that $h_i(x) = 0$ and $\phi_i^{(j)} = 0$. By Theorem 2, we have the initial vector for k-th dimension as $\breve{\Omega}(k) = \Omega(k) + [\phi_1(k), \ldots, \phi_N(k)]^\top$. For simplicity, we denote $\Phi(k) = [\phi_1(k), \ldots, \phi_N(k)]^\top$.

- *Case II: Communication Error*

In the consensus phase, attackers can manipulate values in coefficient vectors. From Theorem 1-2, normal nodes will reach consensus on $\hat{p}_i(K) = [\omega_1^\top \breve{\Omega}(1), \cdots \omega_m^\top \breve{\Omega}(m)]^\top$ while the accurate consensus will be on $p_i(K) = [\frac{\mathbf{1}^\top}{N}\Omega(1), \cdots \frac{\mathbf{1}^\top}{N}\Omega(m)]^\top$. Denote $\delta_i(j) = \hat{p}_i^{(j)}(K) - p_i^{(j)}(K)$ as the communication error on the j-th dimension of node i. Therefore, we provide Theorem 3 to formulate the communication error as follows.

Theorem 3. *Suppose that Assumptions 1-3 hold. With R-ADOA, the communication error* $\delta_i(j)$ *is given by*

$$\delta_i(j) = (\omega_j - \frac{\mathbf{1}}{N})^\top \Omega(j) + \omega_j^\top \Phi(j), \tag{15}$$

where $\forall i \in \mathcal{V}_n$ *and* $\forall 0 \le j \le m$.

Proof. Note that $\delta_i(j) = \hat{p}_i^{(j)}(K) - p_i^{(j)}(K)$. By plugging in the derived formulas of $\hat{p}_i^{(j)}(K)$ and $p_i^{(j)}(K)$, we have

$$\delta_i(j) = \omega_j^\top \breve{\Omega}(j) - \frac{\mathbf{1}^\top}{N}\Omega(j).$$

Next, the formula of $\breve{\Omega}(j)$ is given by $\breve{\Omega}(j) = \Omega(j) + \Phi(j)$ according to Theorem 2. Hence

$$\delta_i(j) = (\omega_j - \frac{\mathbf{1}}{N})^\top \Omega(j) + \omega_j^\top \Phi(j). \qquad \square$$

The communication error contains all errors that happen in the first two phases. The first term is due to errors that malicious nodes impose in the consensus phase. The second term reflects that approximation errors that malicious nodes impose spread to every node. This formula will be further included in the boundary of optimal value.

D. Performance Loss Due to Perturbation

In this subsection, we investigate the sensitivity of optimal value. We only consider the optimal points not locating at the boundary values. In such cases, the optimal points are the roots of the derivative of $p_i^K(x)$. To estimate the sensitivity of roots of this polynomial, an available method is to regard these roots as eigenvalues of the colleague matrix of $\mathrm{d}p_i^K(x)/\mathrm{d}x$ and estimate the perturbation of eigenvalues. First, we illustrate the influence of perturbation on the collage matrix. As illustrated in [26], we have the expression

$$\frac{\mathrm{d}p_i^K(x)}{\mathrm{d}x} = \sum_{j=0}^{m-1} \tilde{p}_i^{(j)}(K)\,T_j\left(\frac{2x-(a+b)}{b-a}\right),\ x \in (a,b),$$

$$\tilde{p}_i^{(j)}(K) = \begin{cases} \tilde{p}_i^{(j+2)}(K) + 2(j+1)Sp_i^{(j+1)}(K), & j = 1, ., m-1, \\ \frac{1}{2}\tilde{p}_i^{(2)}(K) + Sp_i^{(1)}(K), & j = 0, \end{cases} \tag{16}$$

where $S = (b-a)/2$ and $\tilde{p}_i^{(m)}(K) = \tilde{p}_i^{(m+1)}(K) = 0$. Then, we know from [25] that roots of equation $\mathrm{d}p_i^K(x)/\mathrm{d}x = 0$ are eigenvalues of its colleague matrix M_C,

$$\begin{bmatrix} 0 & 1 & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ & \ddots & \ddots & \ddots & \\ \eta_i^0 & \eta_i^1 & \cdots & \frac{1}{2} + \eta_i^{m-3} & \eta_i^{m-2} \end{bmatrix}$$

where $\eta_i^j = -\tilde{p}_i^{(j)}(K)/(2\tilde{p}_i^{(m-1)}(K))$. Since the consensus algorithm is applied coordinate-wisely, there will be error in each dimension. If we assume that $\hat{\tilde{p}}_i^j(K) = \tilde{p}_i^j(K) + \Delta_i(j)$, then from (8) and (9) we can derivate the formula of $\Delta_i(j)$.

From $\tilde{p}_i^{(j)}(K) = \tilde{p}_i^{(j+2)}(K) + 2(j+1)Sp_i^{(j+1)}(K)$ we have that for $l \leq (m-2)/2$

$$\begin{aligned}\tilde{p}_i^{(m-(2l+1))}(K) &= \sum_{t=0}^{l} 2\,(m-2t)\,Sp_i^{(m-2t)}(K),\\ \tilde{p}_i^{(m-2l)}(K) &= \sum_{t=1}^{l} 2\,(m-2t+1)\,Sp_i^{(m-2t+1)}(K).\end{aligned} \tag{17}$$

Directly from equation (17) we derive that

$$\begin{aligned}\Delta_i(m-(2l+1)) &= \sum_{t=0}^{l} 2\,(m-2t)\,S\delta_i\,(m-2t),\\ \Delta_i(m-2l) &= \sum_{t=0}^{l} 2\,(m-2t+1)\,S\delta_i\,(m-2t+1).\end{aligned} \tag{18}$$

Next, we define the relative error for node i at j-th dimension as $\kappa_i(j) = \Delta_i(j)/\tilde{p}_i^j(K)$. Suppose we have perturbed matrix $\hat{M}_C$ as $\hat{M}_C = M_C - E$. Then the formula for entry of E (derived in Appendix C) as below.

$$E_{ij} = \begin{cases} 0, & 0 < \ i \ < \ m-1, \\ \frac{\tilde{p}_i^{(j-1)}(K)}{2\tilde{p}_i^{(m-1)}(K)}\left(1 - \frac{1+\kappa_i(j-1)}{1+\kappa_i(m-1)}\right), & \text{otherwise.} \end{cases} \tag{19}$$

Now we can derive the influence on the optimal point using the eigenvalue perturbation theory of matrices. We apply Lemma 2 to M_C and the following theorem.

Theorem 4. *Suppose Assumptions 1-4 hold and normal nodes adopt R-ADOA. Then, there exists an extreme point $\hat{x}_j$ of disturbed approximation polynomial $\hat{p}_i^K(x)$ such that*

$$|x^* - \hat{x}^*| \leq \left|\hat{x}^* - \hat{x}_j\right| + \max\left\{n_j\theta, \left(n_j\theta\right)^{1/n_j}\right\}, \tag{20}$$

where n_j is the order of $\hat{x}_j$ as a root of $\frac{\mathrm{d}p_i^K(x)}{\mathrm{d}x} = 0$.

Proof. Suppose that x^* is the optimal point. Then, by [25], x^* is also an eigenvalue of its companion matrix. It is from Theorem 4 that there exists an eigenvalue $\hat{x}_j \in \sigma(\hat{M}_C)$ such that $|x^* - \hat{x}_j| \leq \max\left\{n_j\theta, \left(n_j\theta\right)^{1/n_j}\right\}$. As we know that $\hat{x}_j$ is also an extreme point of $\hat{p}_i^K(x)$, the proof is over immediately. □

The above theorem provides an upper bound of the distance between optimal values before and after perturbation. Based on Theorem 4, we obtain an upper bound of $|p_i^K(x^*) - \hat{p}_i^K(\hat{x}_j)|$, i.e., the distance between optimal values of initial function and perturbed distributed approximation function. This bound is given in the following corollary.

Corollary 1. *Suppose that the conditions in Theorem 4 are satisfied. Then, we have*

$$\begin{aligned}\left|f^* - \hat{p}_i^K(\hat{x}^*)\right| \leq \varepsilon_M &+ L\max\left\{n_j\theta, \left(n_j\theta\right)^{1/n_j}\right\}\\ &+ L\left|\hat{x}_j - \hat{x}^*\right| + \sum_{j=0}^{m}\left|\delta_i(j)\right|,\end{aligned}$$

where $\varepsilon_M = \max_{i\in\mathcal{V}_n}\left|f_i(x) - p_i^{(m)}(x)\right|$.

Proof. The proof is going with the following inequality. First, we divide left hand side (LHS) into four parts, i.e.,

$$\begin{aligned}&\left|p_i^K(x^*) - \hat{p}_i^K(\hat{x}^*)\right| \leq \left|f^* - p_i^K(x^*)\right| + \left|p_i^K(x^*) - p_i^K(\hat{x}_j)\right|\\ &+ \left|p_i^K(\hat{x}_j) - \hat{p}_i^K(\hat{x}_j)\right| + \left|\hat{p}_i^K(\hat{x}_j) - \hat{p}_i^K(\hat{x}^*)\right|\end{aligned}$$

Next, the first term follows that

$$\begin{aligned}\left|f^* - p_i^K(x^*)\right| &\leq \frac{1}{n}\sum_{i=1}^{n}\left|f_i(x) - p_i^{(m)}(x)\right|\\ &\leq \frac{1}{n}\sum_{i=1}^{n}\max_{i\in\mathcal{V}_a}\left|f_i(x) - p_i^{(m)}(x)\right|\\ &= \max_{i\in\mathcal{V}_a}\left|f_i(x) - p_i^{(m)}(x)\right| = \varepsilon_M.\end{aligned}$$

The second and last item on the right-hand side is bounded by Lipshitz continuous. The third item can be expanded again by the property of Chebyshev polynomial $|T_i(x)| \leq 1$. Then,

$$LHS \leq \varepsilon_M + L\left|x^* - \hat{x}_j\right| + \sum_{i=0}^{m}\left|\delta_i(j)\right| + L\left|\hat{x}_j - \hat{x}^*\right|.$$

Plug in the upper bound of $x^* - \hat{x}_j$ derived in the proof of Corollary 1 and we get the final boundary as below,

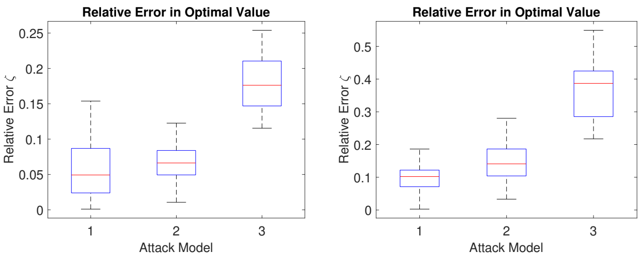
$$LHS \leq \varepsilon_M + L\max\left\{n_j\theta, \left(n_j\theta\right)^{1/n_j}\right\} + L\left|\hat{x}_j - \hat{x}^*\right| + \sum_{j=0}^{m}\left|\delta_i(j)\right|.$$

□

V. Numerical Evaluations

In this section, we run numerical evaluation on a random graph where the probability of edge connection between any two nodes is $p = 0.7$. There are in total 28 normal nodes and 2 adversarial nodes . We design different attacking strategies for two types of attackers and show the performance of R-ADOA on this network. Two kinds of attackers are termed as stochastic and extreme attackers. For stochastic attackers, at each iteration, they can choose values that are bounded by the maximum value and minimum value of normal nodes in each dimension. For extreme attackers, at each iteration, they can choose values that are larger/smaller than the largest/smallest normal nodes, respectively. In Fig. 1, mode 1 denotes stochastic attackers, and mode 2 denotes extreme attackers that take larger values, and mode 3 denotes attackers that take smaller values. In addition, we assume that the error that malicious nodes produce at each coordinate will not exceed the largest coefficient value of all normal nodes, otherwise adversarial nodes will be easily detected.

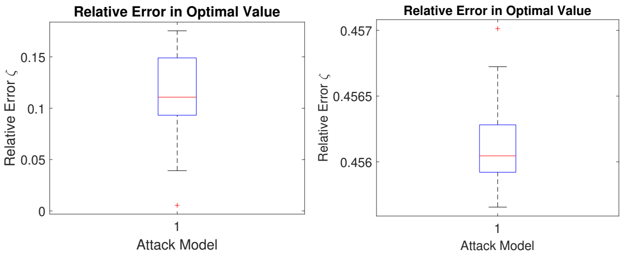
The selection of numerical functions is as below. Parameters are set as $g = 2$ and $n = 30$. For $i = 1, 2, ..., g$, $f_i(x)$ are designed to be adversarial nodes and their Chebyshev proxies are initialized as zero vector. Then for $i = g+1, ..., n$, we set $f_i(x) = \sqrt{1 + (x-i)^2}$. The minimum of objective function is reached at $x = 16.5$ with $f(x) = 7.134$. The approximation order is set to $m = 100$ in the interval $[-30, 30]$. The relative error is used to represent the effectiveness of algorithm and is defined as $\zeta = \left|p_i^K(x^*) - \hat{p}_i^K(\hat{x}^*)\right| / p_i^K(x^*)$. Since the attacker will select value randomly, we repeat the experiment 50 times with each attack model, and the error is plotted in box plots. Similar evaluation is also conducted on $g = 3$ and $n = 40$. The results are shown in Fig. 1.



(a) Relative error on 30 nodes. (b) Relative error on 40 nodes.

Fig. 1: Relative Error of Optimal Value of R-ADOA.

From the picture, we observe that the relative error for attack mode 1 is greatly restricted by R-ADOA. For other cases, the maximum of relative error no larger than 25%. The median relative errors for attack mode 1,2,3 are about 0.05, 0.07 and 0.17 on 30 nodes. The performance of R-ADOA is compared to algorithm in [22] to illustrate the high efficiency and accuracy of R-ADOA. The comparison experiment is conducted with the experiment parameters as $g = 2$, $n = 30$, $p = 0.7$, attack model 1 and same running time. The performance of R-ADOA is shown in Fig. 2a and the algorithm in [22] is shown in Fig. 2b. A straight performance judgement from the Fig. 2 is that R-ADOA is more accurate and Su's algorithm is more stable. One explanation for this contrast is that R-ADOA converges fast, and that by the time R-ADOA finished its computation, Su's algorithm has not reached its end.



(a) Relative error of R-ADOA with 15 iteration rounds. (b) Relative error of [22] with 500 iteration rounds.

Fig. 2: Comparison of R-ADOA with the algorithm in [22].

VI. Conclusion

In this paper, we investigated the safety and robustness of R-ADOA. Unlike traditional resilient distributed optimization algorithms, R-ADOA limits the influence of adversarial nodes to the approximation and the consensus-based iterations. The perturbation of adversaries will ultimately be reflected on the change of coefficients of approximation proxy. After a consensus is reached, each node recovers the approximation polynomial of the global objective function, and the optimization problem turns to find the minimum value of this polynomial. Therefore, the sensitivity analysis of the R-ADOA framework is conducted by studying the influence of perturbation of coefficients on the minimum value of the polynomial. To analyze the sensitivity of the optimal solution, we first estimate the distance between optimal points, and then by using the Lipschitz continuous property we quantify the distance from the optimal value. Future directions include designing more effective filtering mechanisms and considering time-varying directed networks.

Appendix

A. Proof of Theorem 1

The convergence of the W-MSR algorithm is proved in Lemma 1. We first need to prove that when normal nodes reach consensus, the condition formula (12) will be satisfied. Next, we need to show that as long as the formula (12) is satisfied, normal nodes reach consensus.

As R-ADOA adopts W-MSR algorithm in the consensus phase, it is from Lemma 1 that we know the convergence will be reached, which means that $\forall \varepsilon > 0, \exists N_0 \in \mathbb{N}^+,\ \forall t > N_0$, we have $\forall l, k \in \mathcal{V}_n,\ |p_l^{(r)}(t) - p_k^{(r)}(t)| < \varepsilon$. After all normal nodes reaching consensus, our updating rule will only select out values that are the same ohterwise the consensus will not be reached. That is to say $\forall k, l \in \mathcal{N}_i^{(r)*}(t)$ we have $|p_l^{(r)}(t) - p_k^{(r)}(t)| < \varepsilon$. Based on the updating rule of auxiliary variables we know that after the first re-initialization when consensus has been reached, the value of elements in $\mathcal{R}_i^{(j)*}(t)$ and $\mathcal{S}_i^{(j)*}(t)$ are all very close. That is to say that formula (12) will be satisfied.,

Next, we prove that when the formula (12) is satisfied, the consensus has been reached. For any normal node i, we have

$$\begin{aligned}\max_{k,l\in\mathcal{N}_i^{(r)*}(t)} &|p_k^{(r)}(t_o) - p_l^{(r)}(t_o)| \\ &\le |r_i^{(j)}(t_o+1) - \max_{k\in\mathcal{N}_i^{(r)*}(t)} p_k^{(r)}(t_o)| + \delta \\ &+ |s_i^{(j)}(t_o+1) - \min_{l\in\mathcal{N}_i^{(r)*}(t)} p_l^{(r)}(t_o)| \le 3\delta.\end{aligned}$$

Then, for any two nodes u, v, there exists a connected path consisting of only normal nodes from u to v. We denote it as $\mathcal{P}_{uv} = \{x_1, x_2, ..., x_z\}$. Then, we have

$$\begin{aligned}|p_u^{(r)}(t_o) - p_v^{(r)}(t_o)| &\le \sum_{i=1}^{z-1} |p_{x_i}^{(r)}(t_o) - p_{x_{i+1}}^{(r)}(t_o)| \\ &\le \sum_{i=1}^{z-1} \max_{k,l\in\mathcal{N}_i^{(r)*}(t)} \|p_k^{(r)}(t_o) - p_l^{(r)}(t_o)\| \le 3z\delta \le 3D\delta.\end{aligned}$$

As long as δ is smll enough, we have that all normal nodes reach consensus.

B. Proof of Theorem 2

In this proof, the formula of approximation error $\varphi_i(x)$ is derived. We have $f_i(x) = p_i^{(m_i)}(x) + \varepsilon_i(x)$, $\hat{f}_i(x) = \hat{p}_i^{(m_i)}(x) +$

$\hat{\varepsilon}_i(x)$ and $\hat{f}_i(x) = f_i(x) + \varphi_i(x)$. This leads to

$$\begin{aligned}\varphi_i(x) &= \hat{p}_i^{(m_i)}(x) - p_i^{(m_i)}(x) + \hat{\varepsilon}_i(x) - \varepsilon_i(x)\\ &= \sum_{j=0}^{m_i}(\hat{p}_i^{(j)} - p_i^{(j)})T_j\left(\frac{2x-(a+b)}{b-a}\right) + \hat{\varepsilon}_i(x) - \varepsilon_i(x).\end{aligned}$$

Let $t = \frac{2x-(a+b)}{b-a}$ and multiply $\frac{1}{\sqrt{1-t^2}}$ on both sides. Due to the orthogonality of Chebyshev polynomials we have

$$\int_{-1}^{1} T_i(x)T_j(x)\frac{\mathrm{d}x}{\sqrt{1-x^2}} = \begin{cases}0, & \text{if } i \neq j,\\ \pi, & \text{if } i = j = 0,\\ \frac{\pi}{2}, & \text{if } i = j \neq 0.\end{cases}$$

Thus the error ϕ_j at $j \neq 0$-th dimension is as follows

$$\int_{-1}^{1}\frac{\varphi_i\,(x)\,T_j\,(t)}{\sqrt{1-t^2}}\mathrm{d}t = \frac{\pi}{2}\phi_j + \int_{-1}^{1}\frac{(\hat{\varepsilon}_i\,(x) - \varepsilon_i\,(x))\,T_j\,(t)}{\sqrt{1-t^2}}\mathrm{d}t$$

$$\Rightarrow \phi_j = \frac{2}{\pi}\int_{-1}^{1}\frac{h_i\,(t)\,T_j\,(t)}{\sqrt{1-t^2}}\mathrm{d}t,$$

where $h_i\,(t) = \varphi_i\,(x\,(t)) + \varepsilon_i\,(x\,(t)) - \hat{\varepsilon}_i\,(x\,(t))$ and $x(t) = [(b-a)\,t + a + b]\,/2$. Similarly we have for $j = 0$

$$\phi_j = \frac{1}{\pi}\int_{-1}^{1}\frac{h_i\,(t)\,T_j\,(t)}{\sqrt{1-t^2}}\mathrm{d}t.$$

C. Derivation of Formula (19)

In this part, we derive the formula (19) through algebra deformation. From the expression of M_C, it is found that the perturbation on $\tilde{p}_j^i\,(K)$ only affects the last line of M_C. Then, we consider the change of entries in the last line for $i = m-1, 1 \leq j \leq m-1$. The formula is derived as

$$E_{ij} = \frac{\tilde{p}_i^{(j-1)}\,(K)}{2\tilde{p}_i^{(m-1)}\,(K)}. - \frac{\tilde{p}_i^{(j-1)}\,(K) + \Delta_i(j-1)}{2\tilde{p}_i^{(m-1)}\,(K) + 2\Delta_i(m-1)}$$

By reduction of fractions to a common denominator and plug in $\kappa_i(j) = \Delta_i(j)/\tilde{p}_i^j(K)$ we have that,

$$E_{ij} = \frac{\tilde{p}_i^{(j-1)}(K)}{2\tilde{p}_i^{(m-1)}(K)}\left(1 - \frac{1+\kappa_i(j-1)}{1+\kappa_i(m-1)}\right).$$

References

[1] H. Jaleel and J. S. Shamma, "Distributed optimization for robot networks: From real-time convex optimization to game-theoretic self-organization," *Proc. IEEE*, vol. 108, no. 11, pp. 1953–1967, 2020.

[2] M. Chen, N. Shlezinger, H. V. Poor, Y. C. Eldar, and S. Cui, "Communication-efficient federated learning," *Proc. Natl. Acad. Sci. USA*, vol. 118, no. 17, 2021.

[3] C. Zhao, J. He, P. Cheng, and J. Chen, "Consensus-based energy management in smart grid with transmission losses and directed communication," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2049–2061, 2016.

[4] R. Xin, S. Pu, A. Nedić, and U. A. Khan, "A general framework for decentralized optimization with first-order methods," *Proc. IEEE*, vol. 108, no. 11, pp. 1869–1889, 2020.

[5] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annu Rev Control*, vol. 47, pp. 278–305, 2019.

[6] P. Bianchi and J. Jakubowicz, "Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization," *IEEE Trans. Autom. Contr.*, vol. 58, no. 2, pp. 391–405, 2013.

[7] G. Scutari and Y. Sun, "Distributed nonconvex constrained optimization over time-varying digraphs," *Math. Program.*, vol. 176, no. 1, pp. 497–544, 2019.

[8] Z. He, J. He, C. Chen, and X. Guan, "Distributed nonconvex optimization: Gradient-free iterations and globally optimal solution," *arXiv preprint arXiv:2008.00252*, 2020.

[9] T. Tatarenko and B. Touri, "Non-convex distributed optimization," *IEEE Trans. Autom. Contr.*, vol. 62, no. 8, pp. 3744–3757, 2017.

[10] D. Liu and R. Su, "Distributed optimization for linear multi-agent systems subject to dos attacks," in *Proc. IEEE CDC*, 2020, pp. 4498–4503.

[11] Z. Yang, A. Gang, and W. U. Bajwa, "Adversary-resilient distributed and decentralized statistical inference and machine learning: An overview of recent advances under the byzantine threat model," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 146–159, 2020.

[12] C. Zhao, J. He, and J. Chen, "Resilient consensus with mobile detectors against malicious attacks," *IEEE Trans. Signal Process.*, vol. 4, no. 1, pp. 60–69, 2017.

[13] I. Shames, A. M. Teixeira, H. Sandberg, and K. H. Johansson, "Distributed fault detection for interconnected second-order systems," *Automatica*, vol. 47, no. 12, pp. 2757–2764, 2011.

[14] R. Gentz, S. X. Wu, H.-T. Wai, A. Scaglione, and A. Leshem, "Data injection attacks in randomized gossiping," *IEEE Trans. Signal Process.*, vol. 2, no. 4, pp. 523–538, 2016.

[15] L. Yuan and H. Ishii, "Secure consensus with distributed detection via two-hop communication," *Automatica*, vol. 131, p. 109775, 2021.

[16] W. Zheng, Z. He, J. He, and C. Zhao, "Accurate resilient average consensus via detection and compensation," in *Proc. IEEE CDC*, 2021, pp. 5502–5507.

[17] R. M. Kieckhafer and M. H. Azadmanesh, "Reaching approximate agreement with mixed-mode faults," *IEEE Trans. Parallel Distrib. Syst.*, vol. 5, no. 1, pp. 53–63, 1994.

[18] S. M. Dibaji, H. Ishii, and R. Tempo, "Resilient randomized quantized consensus," *IEEE Trans. Autom. Contr.*, vol. 63, no. 8, pp. 2508–2522, 2017.

[19] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 4, pp. 766–781, 2013.

[20] L. Su and N. H. Vaidya, "Fault-tolerant distributed optimization (part iv): Constrained optimization with arbitrary directed networks," *arXiv preprint arXiv:1511.01821*, 2015.

[21] S. Sundaram and B. Gharesifard, "Distributed optimization under adversarial nodes," *IEEE Trans. Autom. Contr.*, vol. 64, no. 3, pp. 1063–1076, 2018.

[22] L. Su and N. H. Vaidya, "Byzantine-resilient multiagent optimization," *IEEE Trans. Autom. Contr.*, vol. 66, no. 5, pp. 2227–2233, 2020.

[23] Z. Yang and W. U. Bajwa, "Byrdie: Byzantine-resilient distributed coordinate descent for decentralized learning," *IEEE Trans. Signal Process.*, vol. 5, no. 4, pp. 611–627, 2019.

[24] K. Kuwaranancharoen, L. Xin, and S. Sundaram, "Byzantine-resilient distributed optimization of multi-dimensional functions," in *Proc. ACC*, 2020, pp. 4399–4404.

[25] L. N. Trefethen, *Approximation Theory and Approximation Practice, Extended Edition*. SIAM, 2019.

[26] J. P. Boyd, *Solving Transcendental Equations: The Chebyshev Polynomial Proxy and Other Numerical Rootfinders, Perturbation Series, and Oracles*. SIAM, 2014, vol. 39.

[27] A. Gil, J. Segura, and N. M. Temme, *Numerical methods for special functions*. SIAM, 2007.

[28] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *IEEE Contr. Syst. Lett.*, vol. 53, no. 1, pp. 65–78, 2004.

[29] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proc. IEEE*, vol. 106, no. 5, pp. 953–976, 2018.

[30] P. Jesus, C. Baquero, and P. S. Almeida, "A survey of distributed data aggregation algorithms," *IEEE Commun. Surv. Tutor.*, vol. 17, no. 1, pp. 381–404, 2014.

[31] V. Yadav and M. V. Salapaka, "Distributed protocol for determining when averaging consensus is reached," in *Proc. 45th Annual Allerton Conf.*, 2007, pp. 715–720.

[32] F. Bullo, *Lectures on Network Systems*, 1st ed. Kindle Direct Publishing, 2021, with contributions by J. Cortes, F. Dorfler, and S. Martinez.

[33] A. Galántai and C. Hegedűs, "Perturbation bounds for polynomials," *Numer Math (Heidelb)*, vol. 109, no. 1, pp. 77–100, 2008.